

## Homework 1: 객체지향 프로그래밍

Due: 03/28

- 제출 요령

- ✓ 프로그램 빌드(build)에 필요 없는 폴더(debug, release, ipch) 및 파일(\*.db)은 모두 삭제
- ✓ 프로젝트 폴더 전체를 ZIP 파일로 압축 후 ClassNet에 제출 (사전 면담 없이 E-mail로 받지 않음)

- 주의 사항

- ✓ Visual Studio 2017만 사용
- ✓ VS 2017에서 솔루션 파일 (\*.sln) 및 프로젝트 파일 (\*.vcxproj)이 open 안되면 0점 처리
- ✓ VS 2017에서 소스 코드(source code)가 빌드 안되면 0점 처리
- ✓ 소스 코드의 copy 적발 시 원본 제공자와 복사자 모두 0점 처리
- ✓ 제출 시간 지나면 0점 처리

- 다음 3가지 기능을 수행하는 콘솔(console) 프로그램을 제작하시오. (10pt)

1. 모든 class의 선언과 정의는 header 파일(\*.h)과 source 파일(\*.cpp)로 분리 (2 pt)
2. Main 함수에서 선택 메뉴(1~4번)를 만들고, 프로그램 종료 때까지 프로그램 선택 메뉴 계속 출력 (1 pt)
  - 1~3번 선택: 각 해당 기능 실행
  - 4번 선택: 프로그램 종료
3. [기능 1] 학생들의 정보를 입력 및 출력하는 CStudent라는 class를 생성하고, 아래와 같이 출력 (2 pt)
  - CStudent class는 학생의 이름(name), 학번(ID), 전공(major), 학년(grade)을 저장 (문자열은 string class를 사용)
  - 학생 정보를 입력 받는 inputData() 멤버 함수 정의 (std::cin 사용)
  - 학생 정보를 출력하는 showData() 멤버 함수 정의 (std::cout 사용)

- 테스트 할 main() 함수 내용:

```
CStudent students[3];

for (int i = 0; i < 3; i++)
    students[i].inputData();

for (int i = 0; i < 3; i++)
    student[i].showData();
```

- 출력 결과 예:

|           |           |        |       |
|-----------|-----------|--------|-------|
| 이름: 홍길동   | 학번: 10010 | 학과: 게임 | 학년: 2 |
| 이름: Davis | 학번: 10011 | 학과: CS | 학년: 1 |
| 이름: 박철수   | 학번: 00012 | 학과: 기계 | 학년: 3 |

4. [기능 2] 시간을 의미하는 CMyTime 이라는 class 를 다음의 조건에 맞도록 정의하고, 주어진 main 함수 내용을 수행했을 때 아래와 같이 출력 (2 pt)

- 시간, 분, 초를 의미하는 정수형 멤버 변수 hour, minute, second를 정의하고, 0으로 초기화
- 시간을 입력 받는 생성자 CMyTime(int h, int m, int s)을 정의 (초와 분의 값 범위는 0~60으로 지정하고, 범위가 넘어가면 다음과 같이 적절하게 값을 조정)

예) 입력값 CMyTime(0, 70, 90) → 저장값 second = 30, minute = 11, hour = 1

- 값을 출력하는 print()라는 멤버 함수를 정의하고, 다음과 같이 항상 각 단위를 두 자리씩 출력  
예) 1시간 3분 4초의 출력시 → 01:03:04
- 시간을 더해 주는 add() 라는 멤버 함수를 정의하고, 입력 인자로 CMyTime class형의 변수를 받음
- 테스트 할 main() 함수 내용:

```
CMyTime a(0, 70, 90);
CMyTime b(0, 0, 3630);

a.print();
b.print();

a.add(b);
a.print();
```

- 출력 결과 예:

```
01:11:30
01:00:30
02:12:00
```

5. [기능 3] 일반적인 도형의 의미하는 CMyShape class와 이를 상속받아 사각형을 의미하는 CMyRect와 원을 의미하는 CMyCircle을 다음 조건에 따라 정의하고, 아래와 같이 출력 (3 pt)

- CMyShape 정의

- ✓ 2차원 위치를 의미하는 멤버 변수: float m\_x, float m\_y (초기화값 0)
- ✓ x, y를 입력 받는 생성자: CMyShape::CMyShape(float fx, float fy)
- ✓ 2차원 위치 정보를 출력해 주는 draw() 멤버 함수  
예) x=1.0, y=2.5 입력 경우 → [MyShape] (1.0, 2.5)

- CMyRect 정의

- ✓ CMyShape에서 상속받음
- ✓ 가로와 세로 크기를 의미하는 멤버 변수: int m\_w, int m\_h (초기값 0)
- ✓ x, y, w, h를 입력 받는 생성자: CMyRect::CMyRect(float fx, float fy, int iw, int ih)
- ✓ 2차원 위치 및 크기 정보를 출력해 주는 draw() 멤버 함수  
예) x=1.5, y=2.0, w=4, h=5 입력 경우 → [MyRect] (1.5, 2.0) w:4 h:5

- CMyCircle 정의

- ✓ CMyShape에서 상속받음
- ✓ 반지름을 의미하는 멤버 변수: float m\_r (초기값 0)
- ✓ x, y, r를 입력 받는 생성자: CMyCircle::CMyCircle(float fx, float fy, float ir)
- ✓ 2차원 위치 및 반지름 정보를 출력해 주는 draw() 멤버 함수  
예) x=1.5, y=2.5, r=7.0일 경우 → [MyCircle] (1.5, 2.5) r:7.0

- 테스트 할 main() 함수 내용:

```
CMyShape * pShape1 = new CMyShape (10.0, 3.5);
CMyShape * pShape2 = new CMyRect (3.0, 5.5, 7, 8);
CMyShape * pShape3 = new CMyCircle (4.5, 5.5, 1.0);

pShape1->draw();
pShape2->draw();
pShape3->draw();

delete pShape1;
delete pShape2;
delete pShape3;
```

- 출력결과:

```
[MyShape] (10, 3.5)
[MyRect] (3, 5.5) w:7 h:8
[MyCircle] (4.5, 5.5) r:1
```